~~Edward G. Poplawski~~

~~Bryan K. Anderson~~
~~Denise L. McKenzie~~
~~Sandra S. Fujiyama~~
~~Olivia M. Kim~~
~~SIDLEY AUSTIN LLP~~
~~555 West Fifth Street~~
~~Los Angeles, CA 90013-1010~~
~~Telephone: (213) 896-6601~~
~~Facsimile: (213) 896-6600~~

~~Attorneys for Plaintiffs/Counterdefendants~~
~~Cornell University and Cornell Research~~
~~Foundation, Inc.~~UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF NEW YORK

CORNELL UNIVERSITY, a non-profit New York corporation, and CORNELL RESEARCH FOUNDATION, INC., a non-profit New York corporation, *Plaintiffs*, v. HEWLETT-PACKARD COMPANY, a Delaware corporation, *Defendant*.	~~Civil Action No.: 01-CV-1974-NAM-DEP~~ ~~PLAINTIFFS'~~**HEWLETT-PACKARD'S PROPOSED REVISIONS TO CORNELL'S PROPOSED CLAIM CONSTRUCTIONS FOR THE JURY** **Civil Action No.: 01-CV-1974-NAM-DEP**
HEWLETT-PACKARD COMPANY, a Delaware corporation, *Counterclaimant*, v. CORNELL UNIVERSITY, a non-profit New York corporation, and CORNELL RESEARCH FOUNDATION, INC., a non-profit New York corporation, *Counterdefendants*.	~~JUDGE~~**Judge**: Hon. Randall R. Rader ~~TRIAL~~**Trial**: May 19, 2008

Defendant Hewlett-Packard Co. ("HP") submits the following proposed claim construction chart to be provided to the jury. For the Court's convenience, a redline comparing Cornell's and HP's proposed charts is attached as Exhibit A.

Claim Constructions

Claim 1:

Term	Court's Definition
"**Concurrencies**"	A plurality of instructions that are ready to be issued because they are dependency free.
"Dispatch stack"	An enriched instruction buffer (or DS) including: (i) A cell for each instruction, each cell having at least the following resultant fields: (1) an operation field (e.g. OP) of the instruction; (2) a field for each source register (one or more e.g. S1) specified by the instruction; (3) an essential dependency field (i.e. α (Si), e.g. α (S1)) corresponding to each source register specified by the instruction; (4) a field for each destination register (one or more e.g. D) specified by the instruction; and (ii) Logic that: (1) decrements a value >0 in the essential dependency field; (2) determines when the value corresponding to α (Si) is zero; and (3) obtains an initial value for α (Si).
"α"	**α is the number of times that a register is used as a destination register in preceding, uncompleted instructions.**
"α(S1)"	**α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions.**
"Source register"	**A register storing an instruction operand.**
"Destination register"	**A register storing an instruction result.**
"Decrement"	**Subtract from the value by the specified amount.**
"Register"	**A data storage element within the processor.**
"Operand"	**A value supplied for an operation performed by a functional unit.**
"Result"	**A value produced by an operation performed by a functional unit.**
"**Execution unit**"	A device containing multiple functional units for executing arithmetic/logic instructions.

Term	Court's Definition
"**Functional unit(s)**"	Performs arithmetic/logic operations on various data types.
"**Instruction**"	An expression that specifies one or more operations and identifies the applicable operands.
"**Instruction issuing system**"	A system comprising an instruction issuing unit.
"**Memory**"	Data storage elements outside the processor for storage of instructions and data.
"**Non-sequential instructions**"	Instructions ordered differently than the order in which they appeared in the instruction stream.
"**Processor**"	A device that interprets and executes instructions.
"**Single processor cycle**"	The shortest amount of time in which a functional unit can complete an operation.

~~Claim 1 (Continued):~~

Means Plus Function Phrases:

"**Means for detecting the existence of concurrencies in said instructions received from said memory**"	The function is "determining the existence of a plurality of instructions that are ready to be issued at the same time because they are data dependency free." The structure corresponding to this function is the "dispatch stack" as defined above for Claim 1. ~~Further, the Court has determined that the PCM is not a necessary structure corresponding to the detecting function~~
"**Means for issuing multiple instructions and non-sequential instructions to said execution unit within a single processor cycle when a concurrency is detected by said means for detecting the existence of concurrencies in said instructions**"	The function is "issuing multiple and non-sequential (i.e., out-of-order) instructions within a single processor cycle that have become dependency free." The structure corresponding to this function is a "reservation circuit." A "reservation circuit" is a common component used with processors and has conventional arbitration logic (i.e., circuitry).

Claim 2:

Term	Court's Definition
"**Arithmetic/logic operation**"	Instructions are sent to their respective functional units along with their respective operands for execution.
"**Concurrencies**"	A plurality of instructions that are ready to be issued because they are dependency free.
"**Dispatch stack**"	An enriched instruction buffer (or DS) including: (i) A cell for each instruction, each cell having at least the following resultant fields: (1) an operation field (*e.g.* OP) of the instruction; (2) a field for each of two source registers (*e.g.* S1 and S2) specified by the instruction; (3) two essential dependency fields (*i.e.* α (S1) and α (S2)) corresponding to the two source registers (*e.g.* S1 and S2) specified by the instruction; (4) a field for each destination register (one or more ~~=~~ *e.g.* D) specified by the instruction; and (ii) Logic that: (1) decrements a value >0 in ~~the~~**each** essential dependency field; (2) determines when the values corresponding to α (S1) and α (S2) are zero; and (3) obtains initial values for α (S1) and α (S2).
"α"	**α is the number of times that a register is used as a destination register in preceding, uncompleted instructions.**
"α(S1)"	**α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions.**
"α(S2)"	**α(S2) is the number of times that an instruction's S2 register is used as a destination register in preceding, uncompleted instructions.**
"Source register"	**A register storing an instruction operand.**
"Destination register"	**A register storing an instruction result.**
"Decrement"	**Subtract from the value by the specified amount.**

Term	Court's Definition
"**Instruction issuing system**"	A system comprising an instruction issuing unit.
"**Memory**"	Data storage elements outside the processor for storage of instructions and data.
"**Operand**"	A value supplied for an operation performed by a functional unit.
"**Register**"	A data storage element within the processor.
"**Result**"	A value produced by an operation performed by a functional unit.

Claim 6:

Term	Court's Definition
"**Arithmetic/logic operation**"	Instructions are sent to their respective functional units along with their respective operands for execution.
"**Concurrencies**"	A plurality of instructions that are ready to be issued because they are dependency free.
"**Dispatch stack**"	An enriched instruction buffer (or DS) including: (i) A cell for each instruction, each cell having at least the following resultant fields: (1) an operation field (*e.g.* OP) of the instruction; (2) a field for each of two source registers (*e.g.* S1 and S2) specified by the instruction; (3) two essential dependency fields (*i.e.* α (S1) and α (S2)) corresponding to the two source registers (*e.g.* S1 and S2) specified by the instruction; (4) a field for each destination register (one or more - *e.g.* D) specified by the instruction; and (ii) Logic that: (1) decrements a value >0 in ~~the~~**each** essential dependency field; (2) determines when the values corresponding to α (S1) and α (S2) are zero; and (3) obtains initial values for α (S1) and α (S2).
"α(S1)"	**α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions.**
"α(S2)"	**α(S2) is the number of times that an instruction's S2 register is used as a destination register in preceding, uncompleted instructions.**
"Source register"	**A register storing an instruction operand.**
"Destination register"	**A register storing an instruction result.**
"Decrement"	**Subtract from the value by the specified amount.**
"**Detecting the existence of concurrencies in instructions stored in said dispatch stack**"	Determining the existence of a plurality of dependency free instructions stored in the dispatch stack.
"**Functional unit(s)**"	Performs arithmetic/logic operations on various data types.
"**Instruction**"	An expression that has a specific format (*i.e.*, OP, S1, S2, D).

Term	Court's Definition
"**Issuing multiple instructions and non-sequential instructions within a given processor cycle when the existence of concurrencies is detected**"	Issuing multiple and non-sequential instructions when the dispatch stack has detected a plurality of concurrently executable (*i.e.* data dependency free) instructions.
"**Issuing instructions**"	Instructions are sent to their respective functional units along with their respective operands for execution.
"Non-sequential instructions"	**Instructions ordered differently than the order in which they appeared in the instruction stream.**
"Operand"	**A value supplied for an operation performed by a functional unit.**
"Processor"	**A device that interprets and executes instructions.**
"Register"	**A data storage element within the processor.**
"Result"	**A value produced by an operation performed by a functional unit.**

~~Claim 6 (Continued):~~

~~Term~~	~~Court's Definition~~
~~"Non-sequential instructions"~~	~~Instructions ordered differently than the order in which they appeared in the instruction stream.~~
~~"Operand"~~	~~A value supplied for an operation performed by a functional unit.~~
~~"Processor"~~	~~A device that interprets and executes instructions.~~
~~"Register"~~	~~A data storage element within the processor.~~
~~"Result"~~	~~A value produced by an operation performed by a functional unit.~~

Claim 14:

Term	Court's Definition
"**A plurality of instructions which are concurrently executable**"	A plurality of instructions that are ready to be issued because they are dependency free.
"**Dispatch stack**"	An enriched instruction buffer (or DS) including: (i) A cell for each instruction, each cell having at least the following resultant fields: (1) an operation field (*e.g.* OP) of the instruction; (2) a field for each source register (one or more *e.g.* S1) specified by the instruction; (3) an essential dependency field (*i.e.* α (Si), *e.g.* α (S1)) corresponding to each source register specified by the instruction; (4) a field for each destination register (one or more *e.g.* D) specified by the instruction; and (ii) Logic that: (1) decrements a value >0 in the essential dependency field; (2) determines when the value corresponding to α (Si) is zero; and (3) obtains an initial value for α (Si).
"α"	**α is the number of times that a register is used as a destination register in preceding, uncompleted instructions.**
"α(S1)"	**α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions.**
"Source register"	**A register storing an instruction operand.**
"Destination register"	**A register storing an instruction result.**
"Decrement"	**Subtract from the value by the specified amount.**
"Register"	**A data storage element within the processor.**
"Operand"	**A value supplied for an operation performed by a functional unit.**
"Result"	**A value produced by an operation performed by a functional unit.**
"**Execution unit**"	A device containing multiple functional units for executing arithmetic/logic instructions.
"**Functional units**"	Performs arithmetic/logic operations on various data types.

Term	Court's Definition
"**Instruction**"	An expression that specifies one or more operations and identifies the applicable operands.
"**Instruction issuing system**"	A system comprising an instruction issuing unit.
"**Memory**"	Data storage elements outside the processor for storage of instructions and data.
"**Non-sequential instructions**"	Instructions ordered differently than the order in which they appeared in the instruction stream.
"**Processor**"	A device that interprets and executes instructions.
"**Single processor cycle**"	The shortest amount of time in which a functional unit can complete an operation.

~~Claim 14 (Continued):~~

Means Plus Function Phrases:

"**Means for detecting the existence of concurrencies in said instructions received from said memory**"	The function is "determining the existence of a plurality of instructions that are ready to be issued at the same time because they are data dependency free." The structure corresponding to this function is the "dispatch stack" as defined above for Claim 14. ~~Further, the Court has determined that the PCM is not a necessary structure corresponding to the detecting function~~
"**Means for issuing multiple instructions and non-sequential instructions to said execution unit within a single processor cycle when concurrently executable instructions are detected by said means for detecting the existence of concurrently executable instructions in said instructions**"	The function is "issuing multiple and non-sequential (i.e., out-of-order) instructions within a single processor cycle that have become dependency free." The structure corresponding to this function is a "reservation circuit." A "reservation circuit" is a common component used with processors and has conventional arbitration logic (i.e., circuitry).

Claim 15:

Term	Court's Definition
"A plurality of instructions which are concurrently executable"	A plurality of instructions that are ready to be issued because they are dependency free.
"**Dispatch stack**"	An enriched instruction buffer (or DS) including: (i) A cell for each instruction, each cell having at least the following resultant fields: (1) an operation field (*e.g.* OP) of the instruction; (2) a field for each source register (one or more *e.g.* S1) specified by the instruction; (3) an essential dependency field (*i.e.* α (Si), *e.g.* α (S1)) corresponding to each source register specified by the instruction; (4) a field for each destination register (one or more *e.g.* D) specified by the instruction; and (ii) Logic that: (1) decrements a value >0 in the essential dependency field; (2) determines when the value corresponding to α (Si) is zero; and (3) obtains an initial value for α (Si).
"α"	**α is the number of times that a register is used as a destination register in preceding, uncompleted instructions.**
"α(S1)"	**α(S1) is the number of times that an instruction's S1 register is used as a destination register in preceding, uncompleted instructions.**
"Source register"	**A register storing an instruction operand.**
"Destination register"	**A register storing an instruction result.**
"Decrement"	**Subtract from the value by the specified amount.**
"Register"	**A data storage element within the processor.**
"Operand"	**A value supplied for an operation performed by a functional unit.**
"Result"	**A value produced by an operation performed by a functional unit.**

Term	Court's Definition
"**Detecting the existence of a plurality of instructions which are concurrently executable from those instructions stored in said dispatch stack**"	Determining the existence of a plurality of dependency free instructions stored in the dispatch stack.
"**Execution unit**"	A device containing multiple functional units for executing arithmetic/logic instructions.
"**Functional unit(s)**"	Performs arithmetic/logic operations on various data types.
"**Instruction**"	An expression that specifies one or more operations and identifies the applicable operands.
"**Issuing instructions**"	Instructions are sent to their respective functional units along with their respective operands for execution.
"Issuing multiple instructions and non-sequential instructions within a given processor cycle when said plurality of concurrently executable instructions are detected"	**Issuing multiple and non-sequential instructions when the dispatch stack has detected a plurality of concurrently executable (*i.e.* data dependency free) instructions.**
"Non-sequential instructions"	**Instructions ordered differently than the order in which they appeared in the instruction stream.**
"Processor"	**A device that interprets and executes instructions.**

Claim 15 (Continued):

Term	Court's Definition
"Issuing multiple instructions and non-sequential instructions within a given processor cycle when said plurality of concurrently executable instructions are detected"	Issuing multiple and non-sequential instructions when the dispatch stack has detected a plurality of concurrently executable (*i.e.* data dependency free) instructions.
"Non-sequential instructions"	Instructions ordered differently than the order in which they appeared in the instruction stream.
"Processor"	A device that interprets and executes instructions.

Claim 18:

Term	Court's Definition
"**Destination register**"	A register storing an instruction result.
"**Register**"	A data storage element within the processor.
"**Source register**"	A register storing an instruction operand.

Authority:

March 26, 2004 Memorandum-Decision and Order ~~(Dkt. No. 225).~~.

Dated: May 9, 2008

Respectfully Submitted,

/s/ Erin Penning
John Allcock (Bar Roll No. 502997)
Sean Cunningham (Bar Roll No. 513394)
Arthur A. Wellman, Jr. (Bar Roll No. 513520)
Erin Penning (Bar Roll No. 513579)
DLA PIPER US LLP
401 B Street, Suite 1700
San Diego, CA 92101-4297
Tel: 619.699.2700 Fax: 619.699.2701
E-mail: john.allcock@dlapiper.com

Barry K. Shelton (Bar Roll No. 503040)
FISH & RICHARDSON P.C.
111 Congress Avenue, Suite 810
Austin, Texas 78701
Tel: 512.226.8105 Fax: 512.320.8935
Email: shelton@fr.com

James C. Moore (Bar Roll No. 102219)
Jerauld E. Brydges (Bar Roll No. 511646)
HARTER, SECREST & EMERY LP
1600 Bausch & Lomb Place
Rochester, NY 14604-2711
Tel: 585.232.6500 Fax: 585.232.2152
E-mail: jbrydges@hselaw.com

Attorneys for Defendant/Counterclaimant
Hewlett-Packard Company